



Paul: Checkpoint Fragen

Checkpointfragen 1

- ▼ Was ist die Aufgabe des Kernels in einem Betriebssystem?

Der Kernel ist die Schnittstelle der Software zur Hardware

- ▼ Wie sähe Programmcode aus wenn es keinen Kernel geben würde?

Es würde nicht funktionieren, da die Schnittschnelle von Soft zu Hardware fehlt.

- ▼ Was ist der Inhalt des Verzeichnisses `/dev/`

Der Inhalt des `/dev`-Verzeichnisses wird auch Geräteknoten genannt.
Geräteknoten repräsentieren Gerätetreiber

- ▼ Was ist der Unterschied zwischen `/dev/sda` und `/dev/sda1`

`sda` ist das komplette Dateisystem und `sda1` die Partition auf dem Dateisystem
`sda`

- ▼ Sie haben Ihren Computer mit einer neuen Hardware verbunden. Wie stellen Sie fest, ob die Hardware erkannt wurde?

Mit dem Befehl `dmesg` kann man die gerade verbundene Hardware überprüfen.

- ▼ Warum exportiert man Umgebungsvariablen?

Umgebungsvariablen werden exportiert damit diese vererbt werden. Das heißt wenn eine Bash in einer Bash gestartet wird, werden diese Variablen in der

neuen Bash dank der Vererbung verfügbar sein.

▼ Von welchem Prozess stammen alle anderen Prozesse in einem Linux-System ab?

Es stammen alle Prozesse von den Eltern Prozess ab. Dieser ist meistens der init Prozess.

▼ Woran erkennt man Kernel-Prozesse?

Kernel Prozesse haben oft eine kurze PID oder die des init-Prozesses, da sie meist direkt nach Start des Systems gestartet werden.

Checkpointfragen 2

▼ Was ist der Unterschied zwischen einem Block- und einem Character-Device?

Block-Devices sind Speicher die Daten in Blöcken senden(Festplatte, USB, CD usw.). Ein Char-Device überträgt Daten zeichenweise (z.B. Tastaturen, Maus usw)

▼ Woran erkennt man, ob ein Gerät ein Block- oder ein Character-Device ist?

Mit dem Befehl `ls -l /dev` werden alle eingehängten Geräte angezeigt. Ein c am Anfang weist auf ein Char-Device hin, ein b hingegen auf ein Block-Device

▼ Was ist das Ergebnis von `mkdir /dev/sda1/ccode` wenn `/dev/sda1` eine Partition einer Festplatte repräsentiert?

Es erscheint eine Fehlermeldung, da es sich hierbei nur um einen Geräteknoten handelt. Im /dev Verzeichnis kann man weder Verzeichnisse anlegen noch Dateien erstellen.

▼ Was wird benötigt, um eine Datei auf der Partition eines Speichermediums anzulegen?

Die Partition muss in einen **Mount-Punkt** eingehängt werden, bevor man etwas anlegen kann.

▼ Wie bekommt man eine Übersicht über alle Block-Devices?

Mit dem Befehl `lsblk` werden alle Block-Devices aufgelistet.

- ▼ Wie kopiert man den Inhalt des Speichermediums /dev/sdc in die Datei /tmp/device.img?

```
sudo dd if=/dev/sdc/ of=/tmp/device.img
```

- ▼ Für welche zwei verschiedenen Zwecke haben wir den Befehl mount genutzt?

Partition der SD-Karte auf Host einhängen & Dateisystem des Targets über NFS auf dem Host einhängen

- ▼ Was ist der Inhalt der Datei /etc/exports?

Hier sind die für NFS freigegebenen Verzeichnisse aufgelistet.

Checkpointfragen 3

- ▼ Mit welchem Befehl erfahren Sie, für welche CPU-Architektur eine ausführbare Datei übersetzt wurde?

```
file Dateiname
```

- ▼ Wofür steht die Abkürzung SoC?

System on a Chip

- ▼ Was ist eine Cross-Toolchain?

Eine Cross-Toolchain ist ein Werkzeugkasten für Software-Entwicklung. Sie erzeugt auf x86 Systemen Binärcode, der auch auf ARM-Systemen läuft.

- ▼ Welche Bedeutung hat die Umgebungsvariable CC?

CC steht für Cross-Compile. CC enthält den Namen des Cross Compilers der Code für ARM Prozessoren erzeugt.

- ▼ Sie finden eine Datei mit der Endung `.so` Was befindet sich in dieser Datei?

`.so` steht für `shared object`, dies sind Dateien die eine Referenz auf die benötigte Bibliothek beinhalten.

- ▼ In einer Datei namens `Makefile` finden Sie die folgenden Zeilen. Was bedeuten Sie?

```
command.o : command.c defs.h command.h
$CC -c command.c
```

Das ist eine Abhängigkeitsregel in einem Makefile, sie enthält ein Target (`command.o`), ein oder mehrere Abhängigkeiten (`command.c defs.h command.h`). `$CC -c command.c` ist ein Kommando um das Ziel zu erstellen.

▼ Sie erhalten die folgende Meldung, als Sie das Programm main ausführen wollen:

```
./main: error while loading shared libraries: libmy.so:
cannot open shared object file: No such file or directory
```

Was ist da los und wie finden Sie die Ursache der Fehlermeldung?

`ldd main` liefert den Grund. Die im Programm verwendete Library wird nicht gefunden.

Checkpointfragen 4

▼ Benennen Sie vier Möglichkeiten, um mit dem Kernel zu kommunizieren.

- Geräteknoten
- Virtuelle Dateisysteme `/sys` oder `/proc`
- System Calls
- Signale

▼ Ihr Home-Verzeichnis enthält sehr viele Dateien, so dass es nicht sinnvoll ist, diese zu zählen. Wie können Sie die Anzahl der Dateien mit Hilfe der Shell bequem ermitteln?

`ls -l /home | wc -l` Findet alle regulären Dateien(-type f) und übergibt diese dann an WordCount. WC listet dann auf, wie viele Lines also Einträge sich darin befinden.

▼ Ist die Funktion `printf()` ein System-Call?

Nein ist sie nicht, jedoch nutzt sie einen Systemcall.

▼ Was ist das Besondere an der Datei kernel7.img ?

Die Datei beinhaltet einen komprimierten Kernel für unser Target.

▼ Der Kernel wird mit Hilfe von make gebaut. Welche Umgebungsvariablen müssen existieren, damit der Bau des Kernels wie erwartet verläuft?

KERNEL=kernel7

▼ Auf der Boot-Partition Ihrer Raspberry Pi gibt es die Datei bcm2709-rpi-2-b.dtb. Was ist der Inhalt der Datei?

Die ist eine Datei mit einer Standardkonfiguration für den Kernel des Raspberry Pi 2B

▼ Man kann aus dem Quellcode des Kernels Standardkonfigurationen erzeugen. Wie finden Sie die möglichen Standardkonfigurationen für den Raspberry Pi?

Auf der Website kernel.org werden Standardkonfigurationen für gängige Plattformen angeboten.

▼ Wo findet man die Konfiguration des Kernels?

In der Konfigurationsdatei des Kernels `.config` im Hauptverzeichnis der Kernelquellen.

▼ Wie finden Sie in einem laufenden Linux System heraus, wann der Kernel gebaut worden ist?

```
uname -a
```

Checkpointfragen 5

▼ Wofür ist das Verzeichnis /usr vorgesehen und was bedeutet sein Name?

Dort befindet sich Software die nicht zur Grundausstattung von UNIX Systemen gehört.

Unix System Resources

▼ Was ist der Unterschied zwischen den Verzeichnissen /mnt und /media?

/mnt = Mountpunkt für Dateisysteme

/media = Mountpunkt für Datenträger, oft automatische Mounts

- ▼ Mit welchem Befehl können Sie den inode einer Datei ermitteln?

```
ls -li
```

- ▼ Mit welchem Befehl erzeugen Sie einen Link namens line.txt auf die Datei read.me?

```
ln line.txt links read.me
```

- ▼ Was ist der Unterschied zwischen einem Hard-Link und einem Soft-Link?

Hardlinks können nicht über Dateisystem-Grenzen hinweg genutzt werden, teilen sich die selbe inode

Softlinks sind symbolische Links die einen eigenen inode bekommen, diese kann man über die Dateisystem Grenzen hinweg benutzen.

- ▼ Was enthält das Array argv in der Parameterliste der main()-Funktion eines C-Programms? Was enthält insbesondere argv[0]?

argv ist ein Argument Array. Das ist ein Array das den Programmnamen und dessen Parameter enthält.

- ▼ Benennen Sie eine Besonderheit der Busybox-Software.

Grundlegende UNIX Befehle in nur einer ausführbaren Datei, Befehle wie ls sind nur Referenzen die an die Busybox weitergeleitet werden.

Checkpointfragen 6

- ▼ Definieren Sie die Begriffe „Architektur“ und „Plattform“, so wie sie in diesem Modul verwendet werden

Architektur = Software-Architektur, Rechner-Architektur und System-Architektur. In unserem Fall beschäftigen wir uns vor allem mit der CPU-Architektur, d.h. Aufbau und Struktur des zentralen Prozessors unseres Rechners.

Plattform = In unserem Fall: CPU Architektur + Hardware-Peripherie + Basis-Software

- ▼ Grenzen Sie Managed-Make- und Makefile-Projekte voneinander ab.

Managed-Make = Hier übernimmt die IDE die Erzeugung, Verwaltung und Anpassung der Makefiles. Probleme gibt es bei Projekten mit mehreren Executables.

Makefile = Ist ein vom Entwickler mitgebrachtes oder erstelltes Makefile und ist dafür auch verantwortlich. Sinnvoll bei z.B. bereits existierenden Projekten.

- ▼ Welche Konzepte werden in Eclipse genutzt, um eine plattformunabhängige Software-Entwicklung effizient zu gewährleisten?

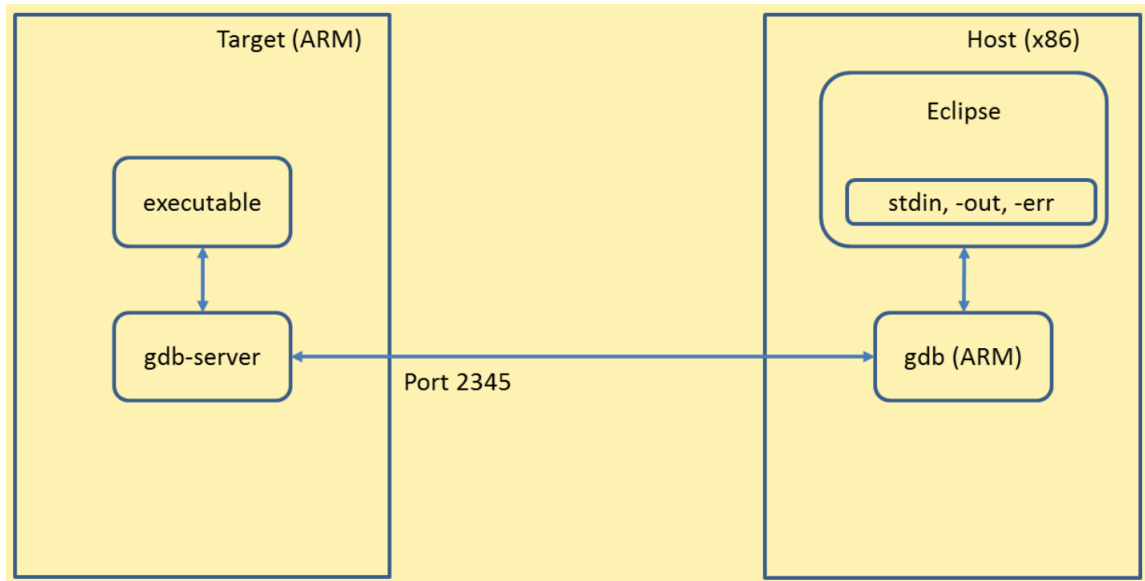
Plattform-Konfiguration = Mit welcher Plattform arbeiten wir gerade?

Build-Konfiguration = Wie soll das Projekt gebaut werden? (z.B. mehrere Executables)

Run-Konfiguration / Remote-Run = Wie soll das Projekt gestartet / ausgeführt werden

Target-Konfiguration = Wie können wir auf ein Target im Netzwerk zugreifen?

- ▼ Skizzieren Sie die beteiligten Instanzen bei einer Remote-Debugging-Session in einem Blockdiagramm.



▼ Eine Datei hat die Permission-Flags 754. Wer hat welche Rechte für diese Datei?

7 - Eigentümer = Lesen, Schreiben, Ausführen

5 - Gruppe = Lesen, Ausführen

4 - Andere Benutzer = Lesen

▼ Welche Klassen von Signalen kennen Sie?

Systemsignale = III, TRAP, BUS

Gerätesignale = HUP, INT, TTIN

Benutzerdefinierte Signale = QUIT, USR1, USR2

▼ Wie sieht das Grundgerüst eines allgemeinen Signal-Handlers aus? Skizzieren Sie in Pseudo-Code.

```

#include <signal.h>

void signalHandler(int signo) {

    if (signalNum == SIGINT) {
        printf "received SIGINT";
    }

    else if (signo == SIGUSR1){
        printf "received SIGUSR1";
    }

}
  
```


▼ Was haben ein Vater- und sein Sohn-Prozeß miteinander gemeinsam?

Was hat ein Main-Thread und die darin erzeugten weiteren Threads miteinander gemeinsam?

Vater und Sohn teilen sich Ressourcen wie Umgebungsvariablen und geöffnete Dateien. Aber der Sohn bekommt seine eigene PID und Speicherbereich.

Ein Main-Thread und die weiteren Threads teilen sich den selben Adressraum und somit die meisten Ressourcen.

▼ Definieren Sie den Begriff „kritischer Bereich“.

Ist ein Bereich bei dem es bei ungünstigen Scheduling zu Inkonsistenzen, d.h zu krassen Fehlern kommen kann.

▼ Was ist der Unterschied zwischen „Deadlock“ und „Livelock“?

Bei einem Deadlock wartet darauf bis jeder Prozess einem anderen Ressourcen freigibt. Keiner der Prozesse kann fortsetzen. Bei einem Livelock sind die Zustände zwar aktiv und ändern Ihre Zustände, aber dabei kein Fortschritt erzielt wird. Die Prozesse reagieren aufeinander, diese Reaktionen hindern das weiter arbeiten.

Checkpointfragen 10

▼ Warum ist Multimedia eine repräsentative Anwendungsdomäne für autonome Systeme?

- Überall vorhanden, breites Einsatzspektrum.
- Auf vielen Plattformen verfügbar
- Leichter Einstieg

▼ Welche Anforderungen implementiert ein modernes Multimedia-Gerät?

Anforderungen: Media Rendering, Media Access, Media Processing, Media Management

▼ Was bedeuten die Begriffe „Bitbreite“ und „Abtaste“ bei der Umwandlung von der analogen in die digitale Repräsentation von Audio-Signalen und welche Auswirkungen haben sie?

Bitbreite: Auflösung der einzelnen Abtastwerte (Samples)

Abtastrate (Sample Rate)=Anzahl Abtastvorgänge pro s.)

Beide haben Auswirkungen auf die Qualität und Größe der Datei.

▼ Was ist das Problem beim Cross-Compiling, wenn das Programm Nicht-Standardbibliotheken nutzt, und wie kann man es – sinnvoll – in Eclipse beheben?

Libs sind zwar da, jedoch nicht im ARM-Format. Cross Compiler (und Cross Linker) kommen nicht weiter. Lösung:

- Gesamtes Root-Filesystem des Targets auf Host einbinden
- #include Pfade des Targets in Eclipse einbinden
- Target Libs beim Linken mit einbinden

▼ Was ist ein Multimedia-Filtergraph? Was ist ein Multimedia-Filter? Was ist ein Multimedia-Framework? **ACHTUNG ANTWORT VON CHAT GPT NICHT IM SKRIPT?!**

Multimedia-Filtergraph: ist eine Darstellung des Datenflusses zwischen verschiedenen Multimedia-Komponenten

Multimedia-Filter: ist eine Komponente, die spezifische Aufgaben wie das Dekodieren oder Filtern von Medieninhalten übernimmt.

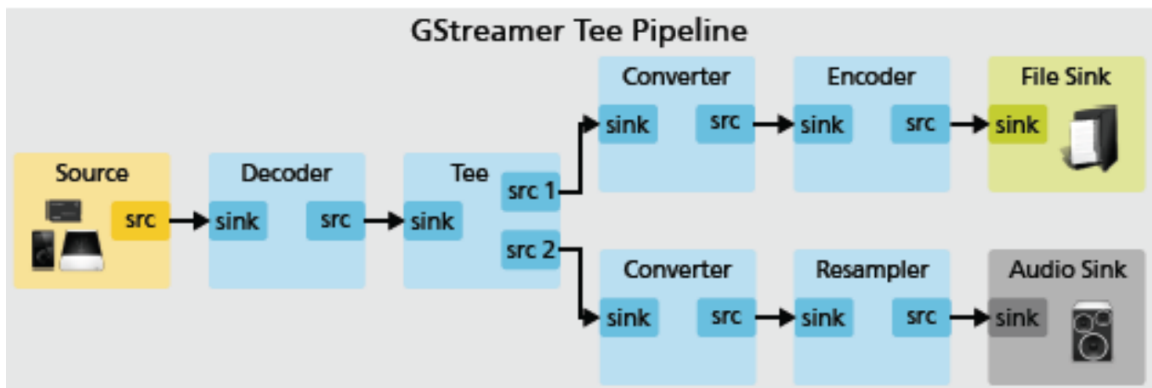
Multimedia-Framework: ist eine Plattform, die Tools und Bibliotheken für die Entwicklung von Multimedia-Anwendungen bereitstellt.

▼ Welche Klassen von Multimedia-Filtern kennen Sie, und welche Aufgaben haben sie?

- Source- / Sink-Filter:
 - Reader, Streamer, Writer (sog. „renderer“)
- Analyze and Control:
 - Parser
 - Demultiplexer
 - Navigator
- Processing:
 - Decoder
 - Encoder
 - Mixer

- A/V Synchronizer
- Queue
- Sample Rate Converter

▼ Skizzieren Sie einen Filtergraphen für einen einfachen Rip&PlayUseCase in der GStreamer-Notation.



Checkpointfragen 11

▼ Was ist ein „Kanal“? Welche Kanäle kennen Sie?

Kanal: alle Werte in einem IMAGE mit demselben Typ

- Farbkanal: für die einzelnen Farben, RGB
- α -Kanal: um die Transparenz zu bestimmen

▼ Was ist Framebuffer, und wie kann in Linux darauf zugreifen?

Speicherbereich, dessen Inhalte den Pixel-Werten des Displays entsprechen

- Anzahl Pixel = Breite w x Höhe h
- Größe [Byte] = Anzahl Pixel x Anzahl Bytes /Pixel

▼ Wie wird ein Pixel im RGB565- und im RGB24-Farbmodell repräsentiert?

RGB565: 16 Bits, also 5 für Rot, 6 für Grün, 5 für Blau. 2 Bytes pro Pixel

RGB24: 24 Bits, für 8 Rot, 8 Grün, 8 Blau. 3 Bytes pro Pixel

▼ Was ist Alphablending und wie wird es erreicht?

Alpha blending ist das Mischen / Überblenden von IMAGES. Ein hoher Wert bedeutet eine geringe Durchsichtigkeit. Dies erreichen wir durch lineare Interpolation zwischen Vordergrundpixel (a) und Hintergrund Pixel (B).

▼ Warum ist ein Framebuffer ein „kritischer Bereich“? Welche Probleme können auftreten und wie werden sie gelöst?

Kritischer Bereich: Bei mehreren Frames die schnell hintereinander erzeugt und angezeigt werden (Animation), greift der Display-Controller auf unvollständige, vermischte und korrupte Inhalte des Framebuffers zu.

Lösung: Double Buffering, also während Buffer A angezeigt wird, dient B zum Erzeugen des nächsten Frames. Dann erfolgt der SWAP und A dient zum Erzeugen des nächsten Frames und B wird gezeigt.

▼ Warum funktioniert der naive Ansatz für Animationen nicht? Wie funktioniert eine einfache Sprite-Animation?

- Beim Naiven Ansatz wird der Hintergrund nicht auf dem vorherigen Zustand zurück gesetzt bevor der nächste Sprite generiert wird. Neben den gewünschten Sprite werden noch die letzten Sprites weiterhin angezeigt.
- Funktionsweise:
 1. Hintergrund Sichern "Save"
 2. Shape zeichnen "draw"
 3. Hintergrund restaurieren "restore"
 4. Shape an neuer Pos. zeichnen